# Scenario smells: signalling potential problems in dialogue scenarios in a serious game

Timo Overbeek[1], Raja Lala[1], and Johan Jeuring[1,2]

[1]*Utrecht University, The Netherlands*
[2]*Computer Science, Open University Netherlands*

**Abstract**

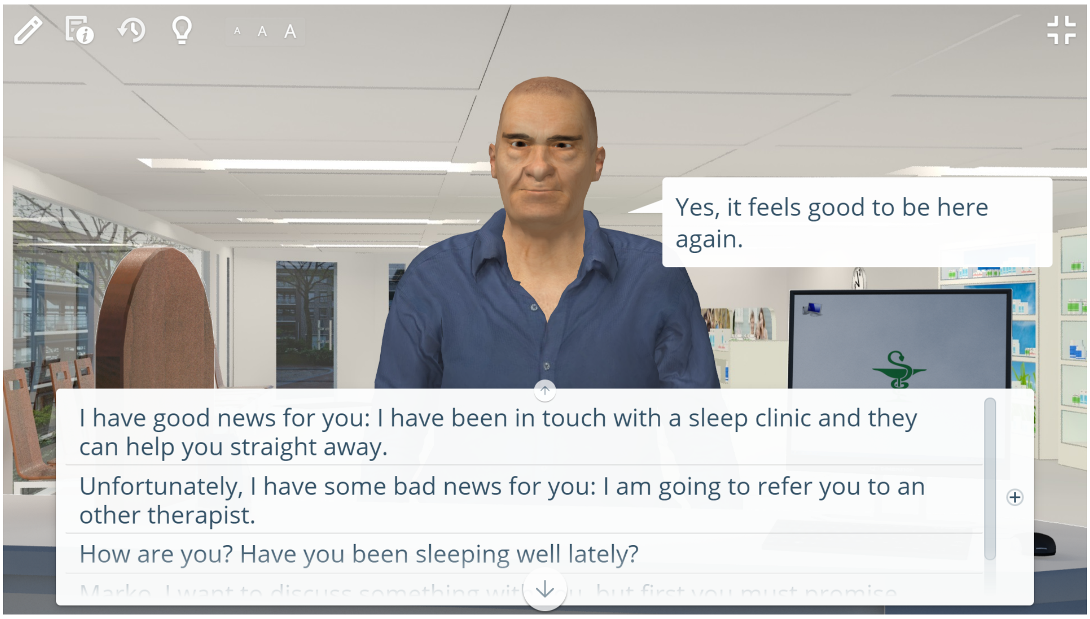
*Many serious games employ a scripted dialogue for player interaction with a virtual character. In our serious game for one-to-one communication skills training Communicate, a scenario author develops a structured, scripted scenario as a sequence of interactions between a player and a virtual character. A player is often a student learning communication skills and a virtual character represents a person that a player talks to, e.g. a patient. A player gets a score on her performance after playing a scenario. Scoring is an important aspect of a scenario. As scenarios get more complex, assigning scores in a scenario also gets more complex, and the risk of an incorrect design increases. To determine what kind of challenges scenario authors experience when assigning scores in a scenario, we conduct a structured interview study with a focus group of scenario authors who use Communicate. Based on these challenges, this paper introduces the concept of scenario smells, and investigates how we can support scenario authors in detecting and addressing such scenario smells. A scenario smell is a symptom of a scenario that could be an indicator of an error or incorrect design in the scenario. We develop a tool that supports a scenario author by identifying scenario smells in a scenario in Communicate. Scenario authors evaluate the tool and find most of the scenario smells useful.*

**Keywords:** Communication skills, scenario smell, serious game, authoring tool

## 1 Introduction

Many serious games offer a player the possibility to perform a dialogue with a virtual character [1–8]. Some of these serious games, such as Communicate [6] and Visual SceneMaker [4], provide a learning environment, where a scenario is created in an authoring tool and thereafter played in a game simulation. A scenario author can create several scenarios in these learning environments. Most of the other serious games offer a single scenario to a player. In all these serious games, a player performs a dialogue with a virtual character (VC) by choosing a statement from multiple statement options, see for an example Fig. 1. Choosing from multiple statement choices resembles a multiple choice test in some ways. After playing a scenario, a player gets a score depending on the choices she made while playing the scenario. A player might play a scenario game simulation several times, trying out different statement options at a step of a scenario, to find out how a VC responds to a particular statement, or to obtain a better score. How do we know that a score given by a game simulation is of good quality? Can we help a scenario author in verifying that the various scores awarded by the game simulation to playing a scenario reflect the intentions of the scenario author?

A scenario in these games is a graph, and a dialogue is a path in this graph of alternating player statement choices and corresponding VC responses. As scenarios become more



Yes, it feels good to be here again.

I have good news for you: I have been in touch with a sleep clinic and they can help you straight away.

Unfortunately, I have some bad news for you: I am going to refer you to an other therapist.

How are you? Have you been sleeping well lately?

Marko, I want to discuss something with you, but first you must promise

**Figure 1:** *Screenshot playing bad news scenario in Communicate*

complex, it becomes more difficult to develop them. One challenging aspect when developing a scenario is to ensure that the scores awarded in a scenario properly reflect the quality of the choices of a player in a scenario. For example, a scenario to practice collaboration skills could have as a goal to balance result- and relation-orientation in raising a difficult issue with a team mate. After playing this scenario a player gets a score percentage on result- and relation-orientation. Ideally, all score combinations are possible (low-low, low-high, high-low, high-high). However, if a player can only obtain low-low or high-high scores in this scenario, the scorings on these aspects are probably correlated. Possibly one of the scoring aspects is superfluous, which is probably not the intention of a scenario author who wants to balance result- and relation-orientation. Correlation of the scoring aspects result- and relation-orientation could be a potential quality problem in this scenario. Another problem might be that it is impossible to obtain high scores on both aspects, in which case a student might consider the assessment in the scenario unfair. Perceived fairness of a multiple choice test affects learning in a student [9, 10]. Correlation of scoring aspects, or the impossibility to achieve high scores on all scoring aspects are indicators of potential quality problems of a scenario. We define a scenario smell as a symptom of a potential scoring problem in a scenario. How can we help a scenario author with detecting and addressing such scenario smells?

This paper addresses the following research questions:

- RQ1: What are the challenges for a scenario author with respect to scoring aspects when developing a scenario?
- RQ2: How can we develop a tool that helps signalling potential problems related to scoring aspects in a scenario?
- RQ3: What are the results of using the tool on actual scenarios?

To investigate scenario smells, and to develop tooling to support signalling them, we need a learning environment that is used by multiple scenario authors to develop their own scenarios, including the scoring of dialogues in the scenario. We choose to investigate scenario smells in the context of the learning environment Communicate. Communicate is widely adopted: more than twenty communication skills teachers (scenario authors) from various universities and faculties use this learning environment in teaching communication skills at several universities. A startup company DialogueTrainer[1] uses it to train their clients. We expect that Communicate is one of most used learning environments for performing dialogues with VCs: the scenario database of Communicate contains more than two thousand (variants of) scenarios, and the amount of users runs in the tens of thousands. Although we answer our research questions in the context of a single learning environment, the results are to some extent generalisable. A scenario in the Communicate learning environment is created in a separate authoring tool, and thereafter played in a game simulation. There is an open source version of the authoring tool[2],and a published scenario language definition. This allows for an independent development of a tool for analysing scenario smells, separate from the Communicate development environment. The open source authoring tool has not only been used to develop scenarios for Communicate, but also for the serious games JobQuest[3] and WaterCooler[4]. WaterCooler is a game where a student learns to develop team building skills. Jobquest is a game where a jobseeker can practice interview skills with a virtual character. For both of these games, game writers developed scenarios using the authoring tool.

To answer RQ1, we conduct structured interviews with six scenario authors using Communicate. These six interviewees include three scenario authors from three different faculties (medicine, pharmacy and veterinary science respectively), one teaching assistant and two scenario authors from the startup company. These interviews lead to formulating several scenario smells. We answer RQ2 by developing a smells tool signalling the scenario smells that the scenario authors deem most important. The smells tool takes a scenario as input and generates a scenario smells report. To answer RQ3, we evaluate the smells tool with four scenario authors using actual scenarios used in practice.

Most of the results of this paper come from an MSc thesis [11]. Some results of the thesis have been published in a conference paper [12], which mainly describes the research problem, and a technical report [13], which describes the initial interviews.

Section 2 describes related work. Section 3 describes the research context, and describes previous work on Communicate. Section 4, 5, 6 answer the research questions RQ1, RQ2, and RQ3, respectively. Section 7 summarises and discusses the answers to the RQ's. Section 8 presents the conclusions and future work.

## 2 Related work

Belotti et al. [14] investigate assessment in serious games. Game play based assessment can provide detailed assessment and be tailored to a player as opposed to a standardised test. Belotti et al. advocate in-game assessment as it does not break a player's game experience. Belotti et al. also find that the proper design of assessment in a game is a challenging and time consuming activity and advocate for supplementary assessment tools.

The concept of providing smells as an indicator of a potential problem to an author (developer) is used in several other disciplines, the most notable being code smells [15] in software development. A code smell is not necessarily an error, but a symptom of a potential quality problem or perhaps a design issue. A code smell is often an indication to investigate if code needs refactoring. An example of a code smell is duplicate code: identical or very similar code that appears in two or more locations in the code base of a piece of software. Another kind of smell are usability smells: Almeida et al. [16] present a catalogue of six usability smells

[1] https://www.dialoguetrainer.com/en/

[2] https://github.com/UURAGE/ScenarioEditor

[3] https://www.gamecomponents.eu/content/598/job-quest

[4] youtu.be/d_irGrEpBdc

as indicators of a user interface (UI) design and maintenance problem. They interview five programmers, and find that most interviewees agree that the proposed usability smells point to potential problems, and that the proposed refactorings lead to improved usability. Almeida et al. also present a tool extension to automate usability smell detection.

Engström et al. [17] describe the challenges faced by a game writer in developing a narrative scenario for a game. Engström et al. mention the limited interest in dialogue/narrative/story writing for games in academia. Game-related academic research often focusses on theoretical reasoning and pedagogical content [18, 19] rather than on more pragmatic game writing. Engström et al. note the absence of a generic tool or format to create narrative scenarios, and mention that a game writer often uses Excel to communicate a dialogue to a game programmer. A game writer sometimes uses a tool to create a prototype of a narrative scenario, and play test it. Engström et al. compare these tools, and find that most tools have a scripting programming interface. They recommend creating a diverse set of tools to support a game writer and present a prototyping tool to support game writers in the domain of point-and-click adventure games.

Some learning environments, such as Enact [7] and deLearyous [20], offer game play tailored to a specific psychological model. Enact [7] is based on five styles of handling interpersonal conflict proposed by Rahim [21]. A scenario author for such a learning environment uses the psychological model for the scenario validation. A scenario in these environments is difficult to adapt to a new psychological model or protocol as opposed to the Communicate learning environment which provides an authoring tool with expressive constructs for easily creating scenarios.

Commercial tools such as for example Chatmapper[5] and Articy[6] offer a visual authoring environment for creating a dialogue. Twine[7] is an open source authoring tool often used for narrative or dialogue writing. Chatmapper uses Lua scripts[8] to set or modify parameter values in a dialogue scenario. Scripting in Lua requires programming skills. Articy is a visual environment for the creation and organization of game content offering virtual characters, locations and a game development environment. Modelling game mechanics, e.g. set or modify parameter(s), requires programming effort. Twine is predominantly aimed at developing a branching narrative and offers scripted programming support, but lacks extensive features to support game mechanics [22]. In the Communicate authoring tool an author has a visual interface to assign conditional logic, set or modify parameter values without the need for programming proficiency. Chatmapper, Articy and Twine offer some form of checking a dialogue. For example, a graph can be checked for completeness and the absence of dead links. This is similar to the Communicate authoring tool, in which a scenario author can check whether all nodes are reachable, nodes do not connect back to a parent, and if every path has an endpoint.

Yessad et al. [23] present an approach to validate game play in an interactive serious game. The approach consists of three steps: specifying a scenario using a generic pattern, model the pattern using a coloured Petri Net, and model checking the coloured Petri Net. Yessad et al. apply this approach to aid the development of a scenario in the design stage of a serious game.

Some game authoring environments generate a 'correct by design' dialogue for a game. Samyn et al. [24] develop a story engine that uses linear scenario templates, metadata, and current game state to generate a scenario. Lessaerd et al. [25] present a tool based on context free grammars (CFG) combined with markup to design a virtual character (VC) dialogue.

[5] www.chatmapper.com

[6] www.nevigo.com

[7] twinery.org

[8] www.lua.org

Lessaerd et al. argue that this tool enables a dialogue author to create new content with predictable runtime VC behaviour, and to identify errors during dialogue creation. However, Lessaerd et al. state that the current version of this tool is not very author friendly, and the proposed benefits have not been tested in practice.

## 3 Research context

In this section, we explain some of the required background knowledge about, and important concepts used in, the description of a scenario in Communicate, since we investigate scenario smells in the context of this learning environment. Almost all of this work has appeared in earlier publications about Communicate.

### 3.1 Authoring a scenario

Usability of an authoring tool often comes at the expense of expressiveness [17, 26]. The open source authoring tool used in Communicate provides expressive constructs for variability in a scenario [27]. A communication skills expert (called scenario author for the rest of the paper), usually a non-programmer, authors a scenario dialogue in the authoring tool. A scenario developed in the authoring tool is validated against an XML schema[9]. A scenario is parsed and produces a scenario 'reasoner' that interacts with the game simulation at run-time to provide information about the possibilities at each step of a scenario. This approach allows a scenario author to develop various scenarios without knowledge of the implementation of the game simulation. The Communicate serious game learning environment already addresses some limitations mentioned by Engström et al. [17], namely offering a graphical interface for authoring a scenario, expressive constructs, a format to validate scenarios and separation of content creation from game simulation.

In this paper, we explore how we can further support a scenario author in developing a scenario, by giving feedback on potential problematic aspects around scoring a scenario. To use an analogy from the Computer Science discipline: we can view the scenario authoring tool as a compiler and the smells tool developed in this paper as a Lint tool.

### 3.2 The structure of a scenario

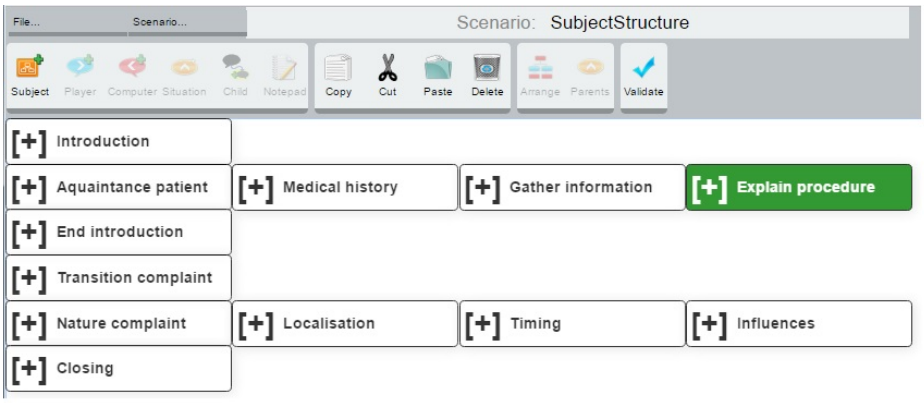
A scenario is a sequence of interleaved subjects (see Fig. 2), where each subject is a directed acyclic graph (see Fig. 3) consisting of a sequence of statements alternating between a virtual character statement and multiple player statement choices.

A scenario author models a communication protocol from top to bottom. A subject usually is a part of a dialogue about a particular subject between a player and a virtual character. If the order in which a player navigates statements across two subjects (or more) is not important, then these subjects may be interleaved, which means that a player can choose between responses that come from both subjects. In the Communicate authoring tool, subjects on the same horizontal level are interleaved. A player gets statement choices from interleaved subjects with no predetermined order in a simulation.

A scenario author defines a learning goal as a parameter in a scenario. An example of such a parameter is 'Empathy'. A parameter is usually encoded as an integer, see Fig. 3.

A player statement (in blue) and a virtual character statement (in red) are nodes and the flow of conversation is an edge. A player statement usually has an incremental score (e.g. Empathy +1), emotional effect on a virtual character (e.g. 'Angry'), and feedback text for a

[9] `https://uudsl.github.io/scenario/`



Scenario: SubjectStructure

[+] Introduction

[+] Aquaintance patient | [+] Medical history | [+] Gather information | [+] Explain procedure

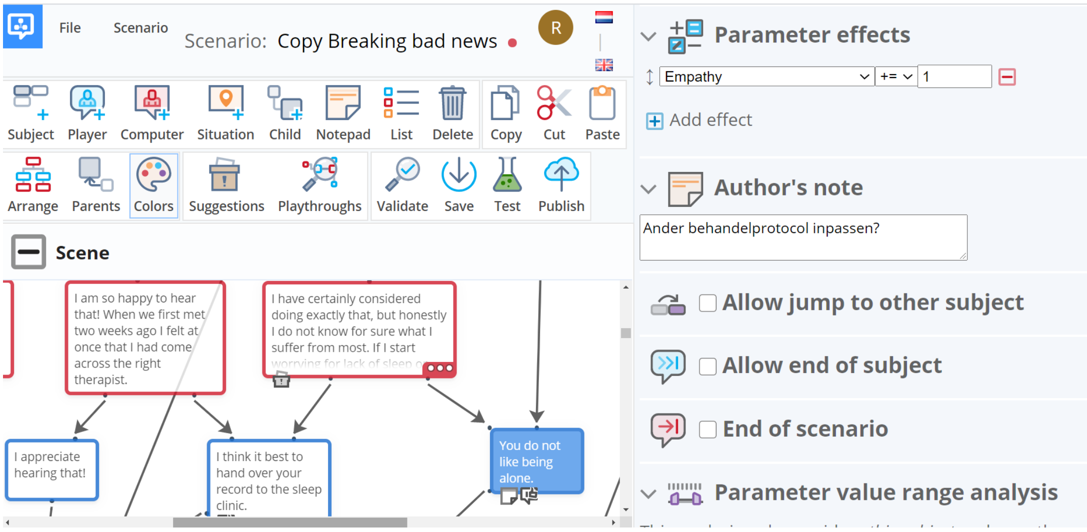
[+] End introduction

[+] Transition complaint

[+] Nature complaint | [+] Localisation | [+] Timing | [+] Influences

[+] Closing

**Figure 2:** *Example interleaving subjects in a Communicate scenario*



Scenario: Copy Breaking bad news

Scene

I am so happy to hear that! When we first met two weeks ago I felt at once that I had come across the right therapist.

I have certainly considered doing exactly that, but honestly I do not know for sure what I suffer from most. If I start worrying for lack of sleep o

I appreciate hearing that!

I think it best to hand over your record to the sleep clinic.

You do not like being alone.

Parameter effects

Empathy += 1

Add effect

Author's note

Ander behandelprotocol inpassen?

□ Allow jump to other subject

□ Allow end of subject

□ End of scenario

Parameter value range analysis

**Figure 3:** *Scenario authoring tool*

player. Multiple player nodes per computer node indicate multiple statement choices for a player. A scenario author assigns a value to a parameter (or multiple parameters) on a player statement choice node.

A scenario author can further increase variability in a dialogue by marking a node with the following:

- Conditional: show only if a particular condition(s) is met.
- Jump: allow to jump from this node to another node in a subject at the same horizontal interleave level.
- Early end of subject: allow to jump from this node to another node in one of the subjects at the same horizontal interleave level. If there are no other subjects at the same level, allow to jump to a node in a subject in the next vertical interleave level.
- End of scenario: terminate a scenario after this statement.

### *3.3 Scoring in a scenario*

Assigning a parameter value across nodes in a graph is challenging and can possibly lead to undesired effects. To illustrate, Fig. 4 shows part of a dialogue with a single parameter in the left graph, while the right graph shows the same dialogue with two parameters. $r_1$ to $r_6$ depict the player statement choice nodes. Note that the statement texts in these nodes is not important.

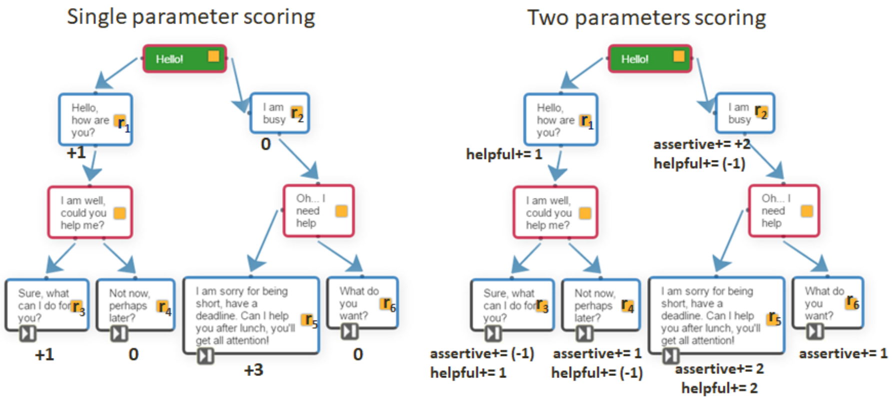

Single parameter scoring
Hello!
Hello, how are you? r1
+1
I am busy r2
0
I am well, could you help me?
Oh... I need help
Sure, what can I do for you? r3
+1
Not now, perhaps later? r4
0
I am sorry for being short, have a deadline. Can I help you after lunch, you'll get all attention! r5
+3
What do you want? r6
0
Two parameters scoring
Hello!
Hello, how are you? r1
helpful+= 1
I am busy r2
assertive+= +2
helpful+= (-1)
I am well, could you help me?
Oh... I need help
Sure, what can I do for you? r3
assertive+= (-1)
helpful+= 1
Not now, perhaps later? r4
assertive+= 1
helpful+= (-1)
I am sorry for being short, have a deadline. Can I help you after lunch, you'll get all attention! r5
assertive+= 2
helpful+= 2
What do you want? r6
assertive+= 1

**Figure 4:** *Scoring challenges example*

When playing the left dialogue, a player can choose between $r_1$ and $r_2$ in the first step of a simulation, where $r_1$ gives a better score. If a player chooses $r_1$, she gets the choices $r_3$ and $r_4$ in the following step of the simulation. It might not be unreasonable to assume that choosing a statement with the highest score at each step results in a total maximum score for a scenario. In this example, the sequence of choices $r_1r_3$ yields a total parameter score of +2. However, the sequence $r_2r_5$ gives a total score of +3, the best score for this parameter. It is possible that the scenario author made an explicit decision for this choice, or might have made an error, but the situation 'smells'.

The use of more parameters increases scoring complexity, as depicted in the right graph in Fig. 4. The best parameter scores are the sequences $r_1r_3$ (assertive: -1, helpful: +2), and $r_2r_5$ (assertive: +4, helpful: +1). Are these parameters (negatively) correlated in this scenario and is that the intention of a scenario author? Scoring complexity further increases with the use of interleaving and other features.

### 3.4 The quality of a scenario

A scenario can be viewed as a multiple-choice test, where the questions posed to the player depend on the answers to the previous questions. When viewed as an assessment, a scenario should be valid and reliable. Besides validity and reliability, there are various other quality aspects of scenarios we can study.

Kasperink [28] applies an argument based approach [29] to validate Communicate and a scenario in Communicate. She validates the 'breaking bad news' scenario for Psychology students, where a student gives bad news to a client. She concludes that the argument based approach is suitable for validating a serious game scenario.

Pel et al. [30] co-create a scenario for shared decision making with co-creation sessions with older adults. Pel et al. evaluate these sessions and improve the scenario by further explaining the goal and the context to positively support the usability and play experience for older people.

Gupta [31] uses psychometric techniques to determine the quality of assessment in a scenario. She analyses the dialogue data from multiple scenarios and determines item difficulty, item discrimination and Cronbach's alpha.

Lala et al. [32] collects student feedback after conducting a communication skills teaching session using Communicate. Lala et al. apply the action research method [33] over three semesters to iteratively improve the quality of a scenario and a communication skills teaching session based on the feedback from students.

In the following section, we introduce scenario smells as another method that can be used by a scenario author to analyse aspects of the quality of a scenario. Scenario smells can be used during scenario development, and do not require player data.

## 4 RQ1: Scenario author challenges and related smells

In this section, we discuss the interviews about challenges in scenario development with six scenario authors, and define various scenario smells based on the results from these interviews.

### 4.1 Challenges for scenario authors in developing scenarios

We interviewed six scenario authors to find out the challenges they face in developing a scenario in Communicate. Our goal was to find information that a scenario author finds important for improving the quality of a scenario. We structured an interview in parts, the entire questionnaire is included in Appendix 9. First, we collected information about a scenario author and how she uses Communicate. Second, we posed questions about how scenario authors develop scenarios. Third, we asked questions related to scenario smells. Fourth, we asked a scenario author what s(he) considers an optimal path in a scenario. Fifth, we proposed enhancements in the authoring tool to aid a scenario author, and asked for open comments. We describe the results of these interviews in the rest of this subsection.

Of the six scenario authors, three were communication skills lecturers at Utrecht University from the faculties of medicine, pharmacy and veterinary science; one was an honours student who recently started using Communicate and two were employees of the startup company DialogueTrainer, which specialises in creating scenarios for training clients in communication skills. The expertise in using the Communicate authoring tool varied considerably, ranging from an author who used only one subject and no conditionality in nodes to an author who used multiple interleaved subjects with entire dialogue paths dependent on a condition.

Most scenario authors created a dialogue first and then assigned parameter and emotional values to a statement. Some scenario authors also created an 'optimal' dialogue path first, and thereafter added ‘erroneous’ paths. The challenges in creating a scenario varied with the experience of the user, for example when asked about the usage of ‘view parents’ and ‘view parameters’ functions, multiple scenario authors indicated that they were not aware of these functions. Scenario authors who did use these functions, used them frequently. Testing a dialogue also varied, some scenario authors played a scenario themselves numerous times, while others let teaching assistants play test a dialogue.

We prepared closed statements to help define scenario smells and asked a scenario author to score the statement on a scale of 1 to 5, where 1 meant complete disagreement with the statement, and 5 meant complete agreement with the statement. We asked scenario authors to rate fifteen statements. Of these fifteen statements, all interviewed scenario authors rated the following with a score of 4 or 5:

- If a player chooses the best option (the option with the highest score increase) at every node, she should have the highest possible score at the end of the scenario.
- It should be possible to get the maximal total score.
- For every parameter there should be a path that generates the maximal score.
- Every node should be reachable.

Scenario authors rated the following statements with a score of 1 or 2 (meaning that they consistently disagree):

- The longest path should yield the highest score.
- The longest path should yield the lowest score.
- Parameters are always positive.

The scenario authors neither agreed nor disagreed with or did not consistently rate the other eight statements (score between 2 and 4):

- The sequence in which subjects have been navigated should not matter for the end score.
- Two parameters should have no correlation.
- It should be possible to get the minimal score.
- It should be possible to get the minimal score on every parameter, but not necessarily the minimal end score.
- The player should sometimes sacrifice a scoring opportunity to receive a better opportunity later.
- A jump point from a subject is not a good idea.
- All parameters should use the same scale (e.g. 1 to 10, 0 to 100).
- It should be possible to score above 50% on every parameter.

We asked the scenario authors seven closed questions on what they considered to be an optimal path in a scenario. One of our goals was to determine if it is possible to unambiguously define the concept of an optimal path in a scenario. The definition most scenario authors prefer is: ‘The optimal path is the path with the highest weighted average of all parameters.’ We refer to the optimal path as the **best score** for the rest of the paper. The **best score** is not a smell as such, but useful for scenario development.

We proposed some potential enhancements to the Communicate authoring environment, and asked for feedback. Three possible enhancements addressed displaying scores in a scenario graph in the Communicate scenario editor. Scenario authors had two primary concerns about these statements: would the proposed enhancements lead to information overload, and how useful is displaying a score in a scenario graph in actual practice? Three possible enhancements related to showing the optimal path while editing. Scenario authors were positive about highlighting an optimal path, even if limited to within a subject. We also asked about the trade-off between calculation time and accuracy. All authors indicated accuracy as more important than calculation time. Three possible enhancements related to marking a particular node as (un)desirable and getting a warning by (not) being on an optimal path; scenario authors were not enthusiastic about these. Finally, two enhancements related to viewing statistics for a scenario. All scenario authors indicated that the maximum score was important; while some authors were interested in the minimum score and the correlation between parameters.

### *4.2 Selected scenario smells*

Based on the interviews, we selected the following scenario smells. We give the name of a smell in brackets; we will use these names in the rest of the paper:

- The minimum (***minimum***) or maximum (***maximum***) score of a parameter is not attainable.
- Always choosing the best option does not result in the maximum score (***always best option***).
- A longest path yields the maximum score (***longest***).
- Two parameters correlate (***parameter correlation***).
- A scenario is not subject monotone (***subject monotonicity***).

Scenario authors assume that a minimum and maximum score of a parameter is attainable while playing a scenario. For example, a scenario author could specify in a scenario definition that a parameter is an integer value between 0 and 10. A player might assume that a score of 6 represents a sufficient result; 8 a good result and 10 a perfect result. If it is impossible to achieve a score higher than an 8 through any path, a ‘perfect’ performance (of 8) corresponds to a score of 80%. If the minimum and maximum score of a parameter is not attainable in a scenario, it constitutes a smell.

Scenario authors indicate that if the longest path yields the maximum score, it is a case of bad scenario design. The reasoning is that a player taking a longer path is perhaps inefficient in a dialogue. If the longest path yields a maximum score, it constitutes a smell.

If two parameters have a positive correlation, perhaps a scenario author should combine these parameters, since having the two different parameters is apparently superfluous. In some cases parameters for conflicting goals, for example as in the right graph of Fig. 4, should have a negative correlation. The sign (positive/negative) and degree of correlation for parameters in a scenario should match the expectations of a scenario author.

On a subject level with interleaving, Communicate allows a player to traverse interleaved subjects in any order. Scenario authors indicate that modifying a scenario can present challenges, especially with interleaved subjects. A dialogue path (good or bad) is difficult to remember. We define a scenario where subjects are independent of each other as a subject monotone scenario. If a scenario is not subject monotone, it represents a smell and we report this to the scenario author.

In addition to the scenario smells indicated by the scenario authors, we develop a ***shortest*** smell. Our assumption is that the shortest path should not yield a maximum score. Most dialogues often have obstacles, such as a bad statement choice that ends a subject or a scenario prematurely. Such a node is often in the shortest path of a scenario. It follows that a shortest path should not result in a maximum score. We define this smell as ***shortest***: A shortest path yields the maximum score for a parameter.

## 5 *RQ2: A tool for reporting scenario smells*

This section describes the rationale of the implementation of our software tool to report scenario smells. We build a tool that takes as input a scenario based on the Communicate scenario language definition[10]. Our tool generates a file that reports the **best score** and scenario smells for the given scenario.

A parameter value in Communicate may decrease in a child node, e.g. for a player statement choice that is a ‘bad’ response. We implement an all paths traversal algorithm, where we compare all paths to determine an optimal path for the **best score**. We use a depth first tree traversal algorithm to find the paths with a maximal score, the paths with a minimal score, the shortest paths, and the longest paths. Further details of the implementation can be found in T.Overbeeks’s Master thesis [11].

We use the Pearson correlation coefficient to calculate correlation between parameters. We calculate the possible end values of each parameter and then apply the Pearson correlation to all pairs of parameters.

Subject monotonicity is based on the mathematical concept of monotonic functions. A monotonically increasing function $f$ is a function for which for all $x$ and $y$, if $x \geq y$, then $f(x) \geq f(y)$. In the case of scoring scenarios, the input consists of the net parameter changes within a subject, and the output is the set of parameter scores at the end of a scenario. A scoring is monotonically increasing if an increase in the net parameter change in a subject leads to an end score set of values equal to or greater than before. A decrease in the net change in one subject should lead to an end value equal to or lower than before for a monotonic decreasing scoring. A scenario that satisfies these conditions is subject monotone. In a subject monotone scenario we consider the subjects to be independent. For instance, if we increase one of the parameter changes in a subject, the end value of that parameter in the scenario will also increase or stay the same.

The output of our tool consists of plain text files. A main file contains an overview of the presence of the different scenario smells.

For example, a portion of the main file looks like:

```
BEST SCORE: 7
FINAL PARAMETER VALUES:
ResultOriented: 8
GoalOriented: 6

Scenario Smells
The longest path yields the maximum score
Correlation between ResultOriented and GoalOriented:
-0.384302256778923
```

[10] https://uudsl.github.io/scenario/

```
GoalOriented
Always choosing the best option does not result in the best score.
```

For detailed information, a scenario author can inspect a text file for each parameter and one for the total score. These files contain path information of the scenario smells for a scenario. Path information consists of a list of the statement texts of the nodes in a path.

## 6 RQ3: Evaluating the scenario smells tool

In this section, we evaluate the scenario smells tool with scenario authors using scenarios used in practice.

We developed the tool in collaboration with a scenario author from Utrecht University, who tested development versions of the software on a scenario he used in blended teaching sessions. This scenario author also used the software tool while making changes to the scenario. We also tested with test scenarios of varying complexity.

The final software was evaluated with four other scenario authors, two from Utrecht University and two from the DialogueTrainer company. One of the DialogueTrainer scenario authors was a new employee and not part of the initial six interviewed scenario authors. The other three scenario authors were part of the initial interviewees. Prior to an interview, a scenario author provided one or two scenarios: an ‘average’ and a complex scenario. The evaluation interview questionnaire is given in appendix 10. We summarise the answers in the following subsection.

### 6.1 Evaluation interviews with four scenario authors

In the initial interviews (Section 4.1), scenario authors indicated the optimal path (**best score**) to be the most important piece of information. In the evaluation round of interviews, scenario authors indicate that the **best score** as a function they will definitely use in practice. In addition to the **best score**, our tool presents the scenario smells listed in Section 4.2. In the evaluation interviews, the scenario authors rate each smell on a scale of 1 to 5, where 1 denotes that they are unlikely to use a smell, and 5 that they will definitely use a smell in practice to improve a scenario. In addition, the scenario authors also provide a rationale for (not) using a smell to improve a scenario.

The top rated smells (score of 4 and above) include: The ***minimum*** and ***maximum*** score of all parameters is not attainable; always choosing the best option (***always best option***) does not result in the maximum score; and a ***longest*** path yields the maximum score. Scenario authors reiterate that a player should experience a scenario as fair and an important aspect is that a highest score is attainable, as also mentioned during the initial interviews.

The smell least likely to be used is if the ***shortest*** path yields the maximum score. The scenario authors indicate a common practice that they include a short erroneous path that often leads to a premature end of a scenario simulation. Since the scenario authors are aware of this erroneous path, they consider this smell unnecessary.

Other lower ranked smells (3.5 and lower) include ***subject monotonicity*** and ***parameter correlation***. Some scenario authors find ***subject monotonicity*** somewhat difficult to understand.

The scenario authors further differentiate between using a scenario smell while developing a scenario and after finishing a scenario. During development, they are more likely to use one or two functions, e.g. ***maximum***, and after finishing a scenario, they might use an entire scenario smells report. While developing a scenario, authors wish to iteratively make changes to a scenario, and run the tool. Thus the tool needs to report scenario smells within a few minutes, or sometimes at most an hour. For correctness after finishing a scenario, scenario authors are willing to wait overnight for a full report.

### *6.2 Tool performance on actual scenarios*

We evaluate our tool performance on a computer with an Intel Core i7 7700 (Quad core, 3.6 GHz). Scenario authors indicate that they use their computer while the tool is running and nowadays most computers have a working memory of 8 to 16 GB. We limit memory usage for the tool to 4 GB for evaluation and the maximum time to execute to a day (24 hours).

Our tool computes a complete scenario smell report for all 'normal' (including test) scenarios well under a minute (max. time 0.5758 mins using 2.2 GB memory). However, four complex scenarios expose the performance limits of the tool. We examine the **best score** and the complete smell report for these four complex scenarios. The scenario characteristics and tool limitations related to these four scenarios are:

- The first complex scenario has four sequential subjects and a highest node count of 72 within a subject. In this scenario, multiple player statement nodes have often no (delta)scores. In our tool, an ***always best option*** tree records all the child nodes with an optimal parameter increase. If no child node increases a parameter value, the search algorithm treats all child nodes equal. A complete report run leads to out of memory within the hardware constraints, the likely cause is the ***always best option*** tree, which is relatively memory intensive. The **best score** calculation runs in 7.74 minutes.

- The second complex scenario has six sequential subjects, node count within a subject ranges from 16 to 58. This scenario is structured so that if a player makes a major mistake during a dialogue, she receives feedback and goes to another extra part of the dialogue. After the initial run fails, we dissect this scenario to test the limits of our tool. We start with only the top level subject, execute our tool and add the next subject and run the tool iteratively. The **best score** run works within our predetermined limits until the 3rd subject (total nodes 154). Adding the fourth subject (total nodes 170) exceeds the 24 hours limit. The complete smell report works only until two subjects (total node count 96) and goes out of memory with the 3rd subject (total nodes 154). The likely cause is the ***always best option*** tree which is memory intensive.

- The third complex scenario has 4 interleaved subjects with six jump points. We analysed that interleaving should have an effect of increasing calculation time by a factorial. We notice an increase in calculation time when adding an interleaved subject in our test scenarios, however this scenario executes within our predetermined limits. The third complex scenario goes out of memory rather than out of calculation time. Analysis of this scenario reveals that numerous child nodes do not have a parameter score and the problem is similar to complex scenario 1. The likely cause is the ***always best option*** tree which treats all child nodes equal in the absence of a 'best option' and is memory intensive.

- Finally, the fourth complex scenario has numerous player statement nodes as a response to a virtual character node. This scenario has four sequential subjects and the node count of the subjects is 92, 65, 78 and 31 respectively. In this scenario, quite a few virtual character nodes have up to eight player choice nodes. In comparison, in most scenarios virtual character nodes have at most three to four subsequent player statement nodes. We predicted that a high path count severely impacts calculation time, but not necessarily memory usage. After the initial run fails, we also dissect this scenario to test the limits. Even with one subject, the calculation time for **best score** is high (759.57

minutes) while the memory usage is relatively low (26 MB). Adding another subject results in an out of calculation time for our tool report.

Summarising, the smells tool processes most scenarios within user-acceptable processing time. These scenarios contain two to eight parameters, one to seven subjects, one to four interleaved (parallel) subjects, and subjects contain 20 to 132 nodes. For a limited number of exceptional scenarios, a combination of these scenario characteristics, e.g. a scenario with more than three interleaved subjects each with more than 60 nodes, leads to a longer than user-acceptable processing time. The ***always best option*** tree is the likely cause in this case. The scenario authors indicated that most scenarios have fewer than three interleaved subjects at any level and that the trend is towards more compact dialogue graphs within a subject. To check these statements, we examined the last ten published scenarios in Communicate from both Utrecht University and the DialogueTrainer company respectively. These twenty scenarios had at most two interleaved subjects at any level and only two scenarios had one subject with more than 60 nodes. We conclude that the smells tool therefore works within user-acceptable processing time for almost all scenarios.

## 7 Discussion

We hypothesized that offering a tool to signal potential scoring issues can support a scenario author. To answer RQ1 (Section 4), we interviewed scenario authors and found that scoring is indeed a difficult aspect of scenario development. Scenario authors consider **best score** as one of the most important aspects of a scenario in Communicate. In addition, we introduced the concept of scenario smells. Signalling scenario smells may support a scenario author by pointing to potentially problematic parts of a scenario.

To answer RQ2 (Section 5), we used tree traversal algorithms in developing a tool to report scenario smells and tested the tool on a number of test scenarios. The smells tool can processes most scenarios in less than a minute.

To answer RQ3 (Section 6), we found that scenario authors were likely to use our tool, using mostly the **best score** during scenario development, and the complete smell report after scenario completion. The smells tool processed most scenarios within user-acceptable processing time. Scenario authors find that a player should experience a scenario as fair and are likely to use smells that help signalling that scoring might be perceived as unfair. Some smells, in particular ***subject monotonicity*** and ***parameter correlation*** were ranked lower than we expected. The interviewees found ***subject monotonicity*** initially difficult to understand, however on explanation were slightly more favourable. ***Parameter correlation*** ratings had a high variance: two authors rated it high, two low. From the initial interviews, we observed that scenario authors had a different approach to scenario development. Different approaches could relate to the likelihood of using (or not using) ***parameter correlation***.

## 8 Conclusion and future work

Communication skills are essential for the majority of professions and are best learned through practice. These skills can be developed in virtual learning environments that simulate real world communication scenarios, for instance a bad news conversation. In Communicate, a scenario author develops a scenario in an authoring tool. A scenario author wants to develop a scenario of good quality, in which the scoring reflects the intentions of the scenario author, and is perceived as fair.

This paper introduced the concept of a scenario smell as a symptom of a potential scoring problem in a scenario. To find out which scenario smells often occur, we determined the scoring challenges that scenario authors face by conducting interviews with scenario authors who use the Communicate authoring tool. To automatically detect the identified scenario smells in a Communicate scenario, we developed a tool that takes a scenario as input and produces a report on the detected smells. We conducted evaluation interviews with scenario authors and evaluated the tool on actual scenarios in Communicate. Scenario authors generally appreciated the feedback they got from the smells tool, and were likely to use it during scenario development. Although we developed and evaluated our work in the context of Communicate, we hypothesize that our results can be applied in any environment that uses scripted dialogues with scoring at each choice a user makes.

In the future we would like to investigate if we can improve the search algorithm, for example by pruning paths with the same parameter scores and investigating the effect on calculation time versus accuracy for a scenario. We could also add a preprocessing step in our scenario smells tool. If an input scenario contains more than three interleaved subjects and/or subject(s) with more than 60 nodes, we can provide a warning message that the processing of this scenario could likely be long. Furthermore, we would like to visualise some scenario smells in the Communicate authoring tool. For example, we could highlight the path to obtain the **best score** in a scenario.

## Availability of data and material

The software tool and tested scenarios are available open source at: `https://git.science.uu.nl/T.J.Overbeek/scenario-smells`

## Acknowledgements

We thank the scenario authors, and J. van Dortmont and M. van Geest from the development team of DialogueTrainer.

This work has been partially funded by the EU H2020 project RAGE (Realising an Applied Gaming Eco-system)[11], Grant agreement no 644187. This paper reflects only the author's view, the European Commission is not responsible for any use that may be made of the information it contains.

## References

[1] T. Bosse and S. Provoost, "Integrating conversation trees and cognitive models within an eca for aggression de-escalation training," in *Proceedings PRIMA 2015: the 18th International Conference on Principles and Practice of Multi-Agent Systems*, ser. LNCS, vol. 9387. Springer, 2015, pp. 650–659, doi: https://doi.org/10.1007/978-3-319-25524-8_48.

[2] L. Bracegirdle and S. Chapman, "Programmable patients: Simulation of consultation skills in a virtual environment," *Bio-algorithms & Med-Systems*, vol. 6, pp. 111–115, 2010.

[3] A. P. Cláudio, M. B. Carmo, V. Pinto, and A. Cavaco, "Virtual humans for training and assessment of self-medication consultation skills in pharmacy students," in *Proceedings*

[11] http://www.rageproject.eu/

*ICCSE 2015: the 10th International Conference on Computer Science & Education*. IEEE, 2015, pp. 175–180, doi: https://doi.org/10.1109/ICCSE.2015.7250238.

[4] P. Gebhard, G. Mehlmann, and M. Kipp, "Visual scenemaker—a tool for authoring interactive virtual characters," *Journal on Multimodal User Interfaces*, vol. 6, no. 1, pp. 3–11, 2011, doi: https://doi.org/10.1007/s12193-011-0077-1.

[5] J. Guo, N. Singer, and R. Bastide, "Design of a serious game in training non-clinical skills for professionals in health care area," in *Proceeding SEGAH 2014: IEEE International Conference on Serious Games and Applications for Health*. IEEE, 2014, pp. 1–6, doi: https://doi.org/10.1109/SeGAH.2014.7067096.

[6] J. Jeuring, F. Grosfeld, B. Heeren, M. Hulsbergen, R. IJntema, V. Jonker, N. Mastenbroek, M. van der Smagt, F. Wijmans, M. Wolters, and H. van Zeijts, "Communicate! — a serious game for communication skills —," in *Proceedings EC-TEL 2015: Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning*, ser. LNCS, vol. 9307. Springer, 2015. ISBN 978-3-319-24258-3 pp. 513–517, doi: https://doi.org/10.1007/978-3-319-24258-3_49.

[7] D. Marocco, D. Pacella, E. Dell'Aquila, and A. Di Ferdinando, "Grounding serious game design on scientific findings: The case of enact on soft skills training and assessment," in *Proceedings EC-TEL 2015 Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning*, ser. LNCS, vol. 9307. Springer International Publishing, 2015. ISBN 978-3-319-24258-3 pp. 441–446, doi: https://doi.org/10.1007/978-3-319-24258-3_37.

[8] L. M. van der Lubbe, C. Gerritsen, D. Formolo, M. Otte, and T. Bosse, "A serious game for training verbal resilience to doorstep scams," in *International Conference on Games and Learning Alliance*. Springer, 2018, pp. 110–120, doi: https://doi.org/10.1007/978-3-030-11548-7_11.

[9] M. E. Martinez, "Cognition and the question of test item format," *Educational Psychologist*, vol. 34, no. 4, pp. 207–218, 1999, doi: https://doi.org/10.1207/s15326985ep3404_2.

[10] P. McCoubrie, "Improving the fairness of multiple-choice questions: a literature review," *Medical teacher*, vol. 26, no. 8, pp. 709–712, 2004, doi: https://doi.org/10.1080/01421590400013495.

[11] T. Overbeek, "Using scenario smells to analyze scripted communication scenarios in virtual learning environments," Master's thesis, Utrecht University, 2019.

[12] R. Lala, J. Jeuring, and T. Overbeek, "Analysing and adapting communication scenarios in virtual learning environments for one-to-one communication skills training," in *iLRN Immersive learning research network conference*, 2017. doi: 10.3217/978-3-85125-530-0-34 Doi: https://doi.org/10.3217/978-3-85125-530-0-34.

[13] T. Overbeek, R. Lala, and J. Jeuring, "Using scenario smells to analyse scripted communication scenarios in virtual learning environments," Department of Information and Computing Sciences, Utrecht University, Tech. Rep., 2017.

[14] F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger, and R. Berta, "Assessment in and of serious games: an overview," *Advances in human-computer interaction*, vol. 2013, 2013, doi: https://doi.org/10.1155/2013/136864.

[15] M. Fowler and K. Beck, *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 1999, doi: https://doi.org/10.1007/3-540-45672-4_31.

[16] D. Almeida, J. C. Campos, J. Saraiva, and J. C. Silva, "Towards a catalog of usability smells," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, 2015, pp. 175–181, doi: https://doi.org/10.1145/2695664.2695670.

[17] H. Engström, J. Brusk, and P. Erlandsson, "Prototyping tools for game writers," *The*

*Computer Games Journal*, vol. 7, no. 3, pp. 153–172, 2018, doi: https://doi.org/10.1007/s40869-018-0062-y.

[18] L. Joyce, "Creating collaborative criteria for agency in interactive narrative game analysis," *The Computer Games Journal*, vol. 4, no. 1-2, pp. 47–58, 2015, doi: https://doi.org/10.1007/s40869-015-0004-x.

[19] H. Koenitz, "Interactive storytelling paradigms and representations: a humanities-based perspective," *Handbook of Digital Games and Entertainment Technologies*, pp. 1–15, 2016, doi: https://doi.org/10.1007/978-981-4560-50-4_58.

[20] J. Wauters, F. Broeckhoven, M. Overveldt, K. Eneman, F. Vaassen, and W. Daelemans, "delearyous: An interactive application for interpersonal communication training," in *Serious Games: The Challenge: Joint Conference of the Interdisciplinary Research Group on Technology, Education, and Communication, and the Scientific Network on Critical and Flexible Thinking Ghent*, ser. Communications in Computer and Information Science, vol. 280. Springer, 2012, pp. 87–90, doi: https://doi.org/10.1007/978-3-642-33814-4_15.

[21] M. A. Rahim, "A measure of styles of handling interpersonal conflict," *Academy of Management journal*, vol. 26, no. 2, pp. 368–376, 1983, doi: https://doi.org/10.5465/255985.

[22] J. Friedhoff, "Untangling twine: A platform study." in *DiGRA conference*, 2013.

[23] A. Yessad, T. Carron, and J.-M. Labat, "An approach to model and validate scenarios of serious games in the design stage," in *International Conference on Web-Based Learning*. Springer, 2013, pp. 264–273, doi: https://doi.org/10.1007/978-3-642-41175-5_27.

[24] K. Samyn, G. Deglorie, P. Lambert, R. Van de Walle, and S. Van Hoecke, "Generating non-linear narrative for serious games with scenario templates," in *Proceedings of the 10th international Conference on computer graphics theory and applications*. SCITEPRESS-Science and Technology Publications, Lda, 2015, pp. 523–530, doi: https://doi.org/10.5220/0005363705230530.

[25] J. Lessard, E. Brunelle-Leclerc, T. Gottschalk, M.-A. Jetté-Léger, O. Prouveur, and C. Tan, "Striving for author-friendly procedural dialogue generation," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*. ACM, 2017, p. 67, doi: https://doi.org/10.1145/3102071.3116219.

[26] T. Murray, "An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art," in *Authoring tools for advanced technology learning environments*. Springer, 2003, pp. 491–544, doi: https://doi.org/10.1007/978-94-017-0819-7_17.

[27] R. Lala, J. Jeuring, J. van Dortmont, and M. van Geest, "Scenarios in virtual learning environments for one-to-one communication skills training," *International Journal of Educational Technology in Higher Education*, vol. 14, no. 1, p. 17, May 2017, doi: https://doi.org/10.1186/s41239-017-0054-1.

[28] S. Kasperink, "Assessing validity of the serious game communicate: An argument-based approach to validation," Master's thesis, Utrecht University, 2017.

[29] M. T. Kane, "An argument-based approach to validity." *Psychological bulletin*, vol. 112, no. 3, p. 527, 1992, doi: https://doi.org/10.1037/0033-2909.112.3.527.

[30] R. Pel-Littel, H. van Zeijts, N. Schram, H. H. Nap, and J. Jeuring, "A training simulation for practicing shared decision making for older patients," *Procedia Computer Science*, vol. 141, pp. 287–293, 2018, doi: https://doi.org/10.1016/j.procs.2018.10.198.

[31] A. Gupta, "Quality of assessment of sequences of choices in the serious game communicate," Master's thesis, Utrecht University, 2019.

[32] R. Lala, G. Corbalan, and J. Jeuring, "Evaluation of interventions in blended learning using a communication skills serious game," in *Games and Learning Alliance*, A. Liapis, G. N. Yannakakis, M. Gentile, and M. Ninaus, Eds. Cham: Springer International

Publishing, 2019. ISBN 978-3-030-34350-7 pp. 322–331, doi: https://doi.org/10.1007/978-3-030-34350-7_31.

[33] K. Lewin, “Action research and minority problems,” *Journal of social issues*, vol. 2, no. 4, pp. 34–46, 1946, doi: https://doi.org/10.1111/j.1540-4560.1946.tb02295.x.

## 9 *Initial interview questionnaire*

### *9..1 General Information*

1. What is your profession and field of expertise?
2. How many scenarios have you created with Communicate?
3. Have you used Communicate in a class room situation?
4. Do you use the validate option?
5. Do you use the view parents option?
6. Do you use the parameter value range analysis?
7. Do you usually work node by node (including an utterance, preconditions, emotions, parameters) or do you first write an entire dialogue and thereafter add parameters and preconditions?

### *9..2 Their Ideas*

8. What kind of problems do you experience in creating Communicate scenarioÕs?
9. Are there problems that are specific to scoring?
10. Do you have any ideas about how we could solve those problems?
11. What kind of steps do you take to avoid problems in scoring?

### *9..3 Validating statements related to possible smells*

I now would like to ask some questions about scenarios. The Communicate authoring tool gives a lot of freedom to develop scenarios. This can be a problem for creating validation tools. We try to make some assumptions about what constitutes a ‘good’ scenario. Could you please describe how true you think the following assumptions are? You have the following options: always true, mostly true, cannot say, mostly false, always false.

12. The longest path yields the maximum score.
13. If a player chooses the best option (the option with the highest score increase) at every node, she should have the highest possible score at the end of the scenario.
14. The sequence in which subjects have been navigated should not matter for the end score.
15. It should be possible to get the maximum total score.
16. It should be possible to get the maximal score on every parameter, but not necessarily the maximum total score.
17. Every node (even those with prerequisites) should be reachable in some way.
18. Two parameters should have no correlation.
19. The longest path should yield the lowest score.
20. It should be possible to get the minimal score.

21. It should be possible to get the minimal score on every parameter, but not necessarily the minimal end score.

22. The player should sometimes sacrifice a scoring opportunity to receive a better opportunity later.

23. A jump point from a subject is not a good idea.

24. All parameters should use the same scale (e.g. 1 to 10, 0 to 100).

25. It should be possible to score above 50% on every parameter.

26. Parameters are always positive.

Could you please select three assumptions that you currently find the hardest to validate in your own scenarios? Are there any assumptions we have missed?

### *9..4 Optimal Path*

An optimal path is a term that usually describes a sequence of nodes that rewards a player with the highest score or the most wins. We could conclude that the path with the highest overall score is the optimal path in the case of Communicate, but this is not necessarily satisfactory. I state some ways to determine the optimal path. Could you please tell me for each definition if you agree or disagree that this would be a workable definition?

27. The path with the highest unweighted average of all the parameters.

28. The path that includes the highest score for an individual parameter.

29. The path with the highest weighted average of scoring parameters. (Note to readers: a scenario author often indicates an total score as a weighted score of parameters)

30. The shortest path.

31. The path with the highest score for a parameter.

32. The longest path.

33. Same as one of the above, but with the extra requirement that all the scores need to be above a certain threshold.

Which of these definitions do you thinks works best? Are there any other possible definitions that we did not think of?

### *9..5 Solutions / UI*

I know discuss possible improvements to Communicate. These are hypothetical improvements, so we do not consider feasibility. For each of these improvements, can you tell me what you think about the following aspects:

- Does using it improves your scenarios?
- Do you think you would make use of this improvement?
- How much margin for errors is there? Should a tool give an exact answer or is it sufficient to give a reasonable answer?

- Would it be okay if we ignore some parts of a scenario in the tool?

34. Visually displaying all the parameter changes for every node.
35. Color coding all the nodes based on the parameter changes. Green for a positive effect on the overall score, orange for no effect and red for a negative effect.
36. Visually displaying the change in the overall score for each node.
37. A button to show the optimal path through a subject.
38. A button to show the optimal path through the entire tree.
39. A button to show the optimal path from a specified node.
40. An option to repeatedly show optimal paths from different start nodes, whereby the start points are randomly chosen by the computer.
41. An option to mark nodes as desirable and to get a warning if a desirable node is not included in the optimal path.
42. An option to mark nodes as undesirable and to get a warning if an undesirable node is included in the optimal path.
43. Statistics about the scores like correlation, minimum value, maximum value etc.
44. A profile of every subject that shows how often a parameter is used in that subject, and what the maximum changes (both positive and negative) are.

### *9..6 Finishing Thoughts*

That was my last question. Now that we have finished the entire interview, do you have any last thoughts. Did you think of any new possible improvements that could be made to Communicate with respect to scoring? Did I miss something important?

# 10 Evaluation interview questionnaire

## 10..1 General Information

1. How many scenarios have you created in Communicate?
2. Are you currently using Communicate in a classroom situation?
3. What is the context of your example scenario?

## 10..2 Demonstration

We will now demonstrate our tool. Afterwards, we want to discuss the different scenario smells. We want to focus on the following aspects:

- Was the result similar to what you expected?
- If not, what was different?
- Would you make changes to this scenario after seeing the results?

We will discuss the scenario smells in this order:

4. Always choosing the best option does not result in the maximum score.
5. A longest path yields the maximum score.
6. A shortest path yields the maximum score.
7. Is it possible to obtain a parameter value lower than the specified minimum value?
8. Is it possible to obtain a parameter value higher than the specified maximum value?
9. Is there a correlation between parameters?
10. Is the scenario not subject monotone?

## 10..3 Our tool

The tool we are about to demonstrate has multiple features. We want to know how useful you think these features are. Can you describe how likely it is you would use these functions by giving a number between 1 and 5, where a 1 means Òwould not use at allÓ, and a 5 means Òwould definitely useÓ?

11. Calculating the best path (overall score).
12. Calculating the worst path (overall score).
13. Calculating the longest path.
14. Calculating the shortest path.
15. Checking if always choosing the best option will lead to the maximum score.
16. Checking if the longest path will also be the best path.
17. Checking if the shortest path will also be the best path.
18. Checking if the minimal and maximum value of a parameter is attainable.
19. Calculating the correlation between different parameters.
20. Checking if a scenario is subject monotone.

### *10..4 Conclusion*

21. Do you think that the demonstrated program might help you improve your scenarios?
22. Are you likely to use the program if it remains separate from the Communicate authoring tool?
23. Are you more likely to use the program if it is integrated in the Communicate authoring tool?
24. Were the calculation times a problem for you?
25. For your personal use, what would be the most useful feature of the current program?